REMARKS

In the above-identified Office Action, the Examiner rejected Claims 1 – 4, 6 – 9, 11 – 14 and 16 - 19 under 35 U.S.C. §102(e) as being anticipated by Nakashima. Claims 5, 10, 15 and 20 were rejected under 35 U.S.C. §103(a) as being unpatentable over Nakashima in view of Potter.

For the reasons stated more fully below, Applicants submit that the claims are allowable over the applied references. Hence, reconsideration, allowance and passage to issue are respectfully requested.

As disclosed in the Responses to the previous Office Actions as well as in the SPECIFICATION, multiprocessor systems are often connected to a network or networks through a limited number (usually one) of physical interfaces. Consequently, before a processor in a multiprocessor system uses a physical interface to transmit network data, it has to first request permission to lock out all the other processors from using the interface. If more than one processor is requesting access to the interface, there may be some access contention or lock contention. To reduce the likelihood of lock contention, an algorithm is generally used to select which one of the requests to honor first. The algorithm may do so on a first-come, first serve or round robin or on a priority basis or using any other contention resolution scheme.

When an access request is honored, the requesting processor is allowed to lock out all other processors from using the interface until the data is transmitted. When the processor has finished transmitting the data, it releases the lock to allow another processor to gain access to the lock. Obviously, while the processor is transmitting data, other processors may issue requests to the lock. Hence, there may be instances when other processors have to wait before gaining access to the physical interface in order to transmit data. In these instances, the physical interface may be viewed as a bottleneck as requests for the physical interface are accumulating at that point.

Thus, although the use of a multiprocessor in a system may greatly improve a computer system's performance, network communications performance may nonetheless not benefit from the use of the multiple processors due to this bottleneck. Therefore, it would be desirable to have a method that alleviates bottlenecks at the physical interface in the point of view of the processors. The present invention provides such a method.

According to the teachings of the invention, when a multiprocessor system that uses a limited number of physical interfaces is to transact data, a determination is made as to whether the data is network data. If the data is network data, the data is transmitted using a virtual Internet protocol (IP) address. The virtual IP address is the IP address of a data holding device rather than the address of a receiving computer.

Thus, the data before being sent onto the network to the receiving computer is sent to the data holding device. This frees up the processors of the multiprocessor system to continue to process data instead of becoming idle, waiting for the data to be transmitted. This may greatly enhance the performance of the multiprocessor system, especially in the case where there is a (long) queue to transmit data onto the network.

The invention is set forth in claims of varying scopes of which Claim 1 is illustrative.

> 1. A method of improving performance in a multiprocessor system that uses a limited number of physical interfaces to transact network data comprising the steps of:
>
> ***determining whether data being processed is network data***; and
>
> ***transacting, if the data is network data, the data using a virtual Internet protocol (IP) address, the virtual IP address being an IP address given to a data holding device in the multiprocessor system***. (Emphasis added.)

The Examiner rejected the independent claims under 35 U.S.C. §102(e) as being anticipated by Nakashima. Applicants respectfully disagree.

Nakashima purports to teach a load-balancing method for a multi-processor system with a plurality of processor modules. In accordance with the purported teachings of Nakashima, the multi-processor system is assigned a virtual IP address while each processor module of the multi-processor system is assigned an IP address. The virtual IP address is used to direct data to the multi-processor system which then determines which one of the processor modules is to handle the data. The determination is made based on the load of each processor module in the system. That is, when a piece of data arrives at the multi-processor system (through the use of the virtual IP address), the system forwards the data to the processor module with the lightest load for processing.

The virtual IP address of the multi-processor system and the IP address of each module are correlated and stored for easy access by a router. This allows the router to send data to the multi-processor system using the virtual IP address when the data contains the IP address of one of the processor modules of the multi-processor system (see paragraph [0028]).

However, Nakashima does not teach, show or so much as suggests the steps of ***determining whether data being processed is network data***; and ***transacting, if the data is network data, the data using a virtual Internet protocol (IP) address***, ***the virtual IP address being an IP address given to a data holding device in the multiprocessor system*** of the claimed invention.

As support for the rejection of the claims, the Examiner stated that Nakashima teaches the determining step of the claimed invention because in the disclosure of Nakashima, "inherently only network data is being transmitted and received between multi-processor information system 1 and router 3 through LAN 2; para 0029." Applicants agree.

But, if only network data is being transmitted, then there is no reason for Nakashima to teach the step of determining whether the data is network data as

the data will always be network data. Consequently, Applicants maintain that Nakashima does not teach the step of ***determining whether data being processed is network data***.

Further, to show that Nakashima teaches the transacting step, the Examiner cited paragraphs 0058 – 0061 of Nakashima. The cited paragraphs state that:

> [0058] As described above, in the present embodiment, assignment of virtual IP addresses can be defined for the processor modules PM#1 to PM#4 on a processor module-by-processor module basis, so that it is possible to determine assignment of a virtual IP address to a processor module depending on the type of a communication process, thereby realizing optimum distribution of loads on the processors.
>
> [0059] More specifically, e.g. in the FIG. 4 example, it is possible to cause all the processors to equally share load of the communication processes.
>
> [0060] In the FIG. 5 example, by assigning virtual IP addresses on a process-by-process basis, it is possible to assign processor modules to the processes, on a process-by-process basis. For instance, it becomes possible to assign the virtual IP address VIPA#1 to processes demanded by ordinary users and assign the virtual IP address VIPA#2 to those demanded by important customers.
>
> [0061] In the FIG. 6 example, for instance, the virtual IP address VIPA#1 can be assigned to processes related to Internet Protocol, and the processor modules PM#3 and PM#4 other than the processor modules PM#1 and PM#2 assigned with the virtual IP

> address VIPA#1 can be assigned to processes related to other protocols, e.g. OSI (Open Systems Interconnection). Thus, by making the processor modules specialized on a protocol-by-protocol basis, the speed of the communication processing can be enhanced.

Thus in the cited paragraphs, it is disclosed that the multi-processor system may have more than one virtual IP address and that the correlation of a real IP address of a processor module to a virtual IP address of the system may be based on communication processes (paragraphs [0058] and [0059]), users (paragraph [0060]), or protocols (paragraph [0061]). But nowhere is there a teaching that ***if the data is network data, the data is transacted using a virtual Internet protocol (IP) address***, where ***the virtual IP address is an IP address given to a DATA HOLDING DEVICE in the multiprocessor system***. Rather, Nakashima teaches that the virtual IP addresses and indeed IP addresses in general are assigned to the processor modules and the system.

Therefore, Applicants again maintain that Nakashima does not teach the step of ***transacting, if the data is network data, the data using a virtual Internet protocol (IP) address***, ***the virtual IP address being an IP address given to a data holding device in the multiprocessor system***.

In rejecting Claim 2 and its counterpart claims (i.e., Claims 7, 12 and 17), the Examiner stated that "Nakashima also discloses the data holding device is a buffer (e.g., storage means 1e, storage device 10g, or shared memory 10h; para 0033, and 0049-0050)." Applicants submit that Nakashima does not teach the limitations of the claims.

Firstly, a buffer, as is well known in the field, is a temporary storage area, usually in RAM whose purpose is to act as a holding area, enabling a CPU to manipulate data before transferring it to a device. Nowhere in the paragraphs cited by the Examiner does Nakashima state, impliedly or otherwise, that the storage means 1e, storage device 10g or shared memory 10h is a buffer. Nakashima merely states that the storage means 1e is used to store the

correlated virtual IP addresses (paragraphs [0027] and [0033]), the storage device 10g is used to store programs executed by the processor modules (paragraph [0049]) and the shared memory 10h is a RAM (paragraph [0050]).

Secondly, based on Claim 1 on which Claim 2 depends, the virtual IP address is assigned to the buffer. Nakashima does not teach, show or suggests that virtual addresses are to be assigned to a buffer as in the claimed invention.

Therefore, Applicants submit that Claim 2 is not anticipated by the teachings of Nakashima.

Once again, the Examiner used paragraphs [0033], [0049] and [0050] to reject a set of claims. In this particular case, in rejecting Claim 3 and its counterpart claims (i.e., Claims 8, 13 and 18) the Examiner stated that "Nakashima also discloses the buffer (1e, 10g 10h) is implemented using memory allocation." Applicants once again disagree with the Examiner's statement.

As mentioned above, Nakashima never teaches that storage means 1e, storage device 10g, and shared memory 10h are buffers. Nakashima merely states that (1) the storage means 1e is used to store the virtual IP addresses (se paragraph [0027]), (2) the storage device 10g is a magnetic storage device (see paragraph [0049]), and (3) the shared memory 10h is a memory the use of which is shared by the processor modules 10a to 10d, and is formed by a RAM (see paragraph [0050]).

Since Nakashima never mentions that the storage means 1e, the storage device 10g, and the shared memory 10h are implemented using memory allocation, Applicants submit that these claims too are not anticipated by the teachings of Nakashima.

Regarding the rejection of Claim 4 and its counterpart claims (i.e., Claims 9, 14 and 19), the Examiner stated that "Nakashima also discloses the buffer contends for access to one of the limited physical interfaces..." in paragraphs [0010] and [0011]. Applicants respectfully disagree.

The claims in the present invention are directed toward improving performance of ***a multiprocessor system that uses a limited number of physical interfaces*** to transact network data. To improve performance, data to be transmitted is sent to a buffer that has been assigned a virtual IP address. This allows the processors to continue to process other pieces of data instead of holding on unto the data to be transacted. Since, there may be more than one buffer holding data to be transacted, they will have to contend to the limited number of physical interfaces, which is the limitations of the claims in question.

Paragraphs [0010] and [0011] state:

> [0010] The present invention has been made in view of the above problems, and an object of the invention is to provide a multiple-processor information processing system which is capable of reducing adverse influence of overload of a predetermined communication process on other communication processes than the predetermined communication process, when the system uses a virtual IP address.
>
> [0011] To attain the above object, the present invention provides a multiple-processor information processing system including a plurality of processor modules. This multiple-processor information processing system comprises virtual IP address definition means for defining virtual IP addresses on a processor module-by-processor module basis, storage means for storing virtual IP addresses defined by the virtual IP address definition means and information indicative of ones of the processor modules corresponding to the virtual IP addresses, respectively, in a state correlated with each other, and notification means for notifying a router of a virtual IP address of each processor module and a real IP address of the each process module as routing information, for

> the each processor module having the virtual IP address stored in the storage means.

Nowhere in the reproduced paragraph above does Nakashima so much as suggest the claimed limitations. Therefore, Applicants submit that, just as in the case of the preceding claims, Claim 4 is not anticipated by the teachings of Nakashima.

Claim 5 and counterpart Claims 10, 15 and 20 were rejected as being unpatentable over Nakashima in view of Potter. Notwithstanding the arguments above regarding Claims 1 – 4 not being anticipated by Nakashima, Applicants submit that the instant claims are patentable over the applied references.

Potter purports to teach a Dynamic addressing mapping to eliminate memory resource contention in a symmetric multiprocessor system. The address mapping technique defines two logical-to-physical address mapping modes that may be simultaneously provided to the processors of the arrayed processing engine to thereby present a single contiguous address space for accessing individual memory locations, as well as memory strings, within the memory resources. These addressing modes include a bank select mode and a stream mode.

The bank select mode uses high-order address bits to select a bank of a memory resource for access. A data structure, such as a table having relatively short entries, is placed within a single bank of memory and addressed using the bank select mode. Thus, a first processor in a first pipeline of the arrayed processing engine can access a random location within this table at absolute time N. As long as the skew between pipelines is as large as the time that the bank is tied up for a single access (e.g., 7 cycles), a second processor in the same column of a second pipeline can execute the same instructions (skewed by the 7 cycles). In this case, the second processor may access the same or a different location within the table (and bank) at time N+7 without contending with the first processor.

The stream mode uses low-order address bits to select a bank within a memory resource. Here, the data structure is preferably a table having relatively long entries, each containing words that are accessed over a plurality of cycles. Accordingly, the long entries are spread across successive banks and stream mode addressing functions to map each successive word to a different bank. By defining the table entry width as a multiple of the access width times the number of banks, contentions can be eliminated. That is, a processor of a first pipeline can access a first word of a random entry from a table resident in Bank 0 at absolute time N; that processor may then access a second word of the same entry from Bank 1 at time N+7. This process may continue with the processor "seeing" the entire entry as a contiguous address space. A corresponding processor of a next pipeline is skewed by 7 cycles and can execute the same instructions for accessing the same or different entry from the same table. Here, a first word is accessed from Bank 0 at time N+7, a second word is accessed from Bank 1 at time N+14, etc., without contention.

However, Potter does not teach, show or suggest the limitations of transmitting the data to the physical interface, the virtual IP address is replaced by a destination IP address.

The Examiner stated that in col. 5, lines 49 – 65 Potter does teach those limitations. Applicants disagree.

In col. 5, lines 49 – 65 it is stated:

> A routing processor 260 executes conventional routing protocols for communication directly with the processing engine 300. The routing protocols generally comprise topological information exchanges between intermediate stations to determine preferred paths through the network based on, e.g., destination IP addresses. These protocols provide information used by the processor 260 to create and maintain routing tables. The tables are loaded into the external partitioned memories 400 as forwarding information base (FIB) tables used by the processing engine to perform forwarding

> operations. When processing a header in accordance with IP switching, the engine 300 determines where to send the packet by indexing into the FIB using an IP address of the header. Execution of the forwarding operations results in destination media access control (MAC) addresses of the headers being rewritten by the processing engine to identify output ports for the packets.

But note that Potter does not teach that a ***virtual IP address is replaced by a destination IP address*** when data is transmitted to a physical interface.

Based on the foregoing, Applicants submit that the claims in the Application are patentable over the applied references. Consequently, Applicants once more respectfully request reconsideration, allowance and passage to issue of the claims in the application.

Respectfully Submitted

By: _____
Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969